

- Altibase 7.1.0.3.8 Patch Notes
  - Fixed Bugs
    - BUG-47681 core 갯수가 129개 이상인 장비에서 Disk 테이블의 dml 동시성 테스트 중, The tablespace does not have enough free space ( TBS Name :SYS\_TBS\_DISK\_UNDO ). 오류가 발생합니다.
    - BUG-47787 recursive with 구문이 중첩되고 CASE WHEN의 Subquery로 사용될 경우 비정상 종료 할 수 있습니다.
    - BUG-47806 sm cursor interface 에서 인자 정합성 검사.
    - BUG-47807 계층형 쿼리에 join을 같이 사용하고 start with구문에 level 컬럼이 사용된 경우 비정상 종료할 수 있습니다.
    - BUG-47813 PCB를 LRU List에 등록한후 PCB에 접근하면 동시성 문제가 발생할 수 있습니다.
    - BUG-47820 start with 구문에 level pseudo column이 사용될 경우 결과값이 틀립니다.
  - Changes
    - Version Info
    - 호환성
    - 프로퍼티
    - 성능 뷰

# Altibase 7.1.0.3.8 Patch Notes

## Fixed Bugs

**BUG-47681 core 갯수가 129개 이상인 장비에서 Disk 테이블의 dml 동시성 테스트 중, The tablespace does not have enough free space ( TBS Name :SYS\_TBS\_DISK\_UNDO ). 오류가 발생합니다.**

- **module** : sm-disk-page
  - **Category** : Efficiency
  - **재현 빈도** : Always
  - **증상** : core 갯수가 129 이상의 경우 TRANSACTION\_SEGMENT\_COUNT 프로퍼티(기본값 265)가 무시되고 512로 설정되어 발생하는 문제입니다. TRANSACTION\_SEGMENT\_COUNT 의 기본값인 256으로 설정되도록 변경합니다.
- 이 패치의 영향으로 129 core 갯수가 129개 이상인 장비에서는 TRANSACTION\_SEGMENT\_COUNT 프로퍼티가 256으로 변경되니, 기존과 동일하게 유지하려

면 해당 프로퍼티의 값은 512로 변경해주어야 합니다.

- 재현 방법
  - 재현 절차
  - 수행 결과
  - 예상 결과
- Workaround

```
SCALABILITY_PER_CPU = 1
```

- 변경사항
  - Performance view
  - Property
  - Compile Option
  - Error Code

## **BUG-47787 recursive with 구문이 중첩되고 CASE WHEN의 Subquery로 사용될 경우 비정상 종료 할 수 있습니다.**

- **module** : qp
- **Category** : Fatal
- **재현 빈도** : Always
- **증상** : recursive with 구문이 중첩되고 CASE WHEN의 Subquery로 사용될 경우 비정상 종료 하는 문제를 수정합니다.
- **재현 방법**
  - 재현 절차

```

drop table CA_ACGRP_LDTL;
drop table CA_COA_MST;
drop table CA_ACGRP_DTL;
drop table CA_ACGRP_MST;
drop table CA_COA_LDTL;
create table "CA_ACGRP_LDTL"
(
    "GAAP_CD" VARCHAR(20) fixed not null,
    "FORM_CD" VARCHAR(80) variable not null,
    "ACGRP_CD" VARCHAR(80) variable not null,
    "LANG_CD" VARCHAR(20) fixed not null,
    "ACGRP_NM" VARCHAR(400) variable not null
);
create table "CA_COA_MST"
(
    "GAAP_CD" VARCHAR(20) fixed not null,
    "ACCT_CD" VARCHAR(80) variable not null,
    "ACCT_NM" VARCHAR(400) variable not null,
    "FULL_ACCT_NM" VARCHAR(800) variable,
    "DRCR_CD" VARCHAR(20) fixed not null,
    "ACCT_LV" NUMERIC(3),
    "ROOT_ACCT_CD" VARCHAR(80) variable,
    "UP_ACCT_CD" VARCHAR(80) variable,
    "FILL_YN" VARCHAR(4) fixed default 'N' not null,
    "ICTR_TP" VARCHAR(20) fixed,
    "ACCT_FG" VARCHAR(12) fixed default '11' not null,
    "PRINT_NO" VARCHAR(160) variable default '1' not null
);
create table "CA_ACGRP_DTL"
(
    "GAAP_CD" VARCHAR(20) fixed not null,
    "FORM_CD" VARCHAR(80) variable not null,
    "ACGRP_CD" VARCHAR(80) variable not null,
    "ACCT_CD" VARCHAR(80) variable not null
);
create table "CA_ACGRP_MST"
(
    "GAAP_CD" VARCHAR(20) fixed not null,
    "FORM_CD" VARCHAR(80) variable not null,
    "ACGRP_CD" VARCHAR(80) variable not null,
    "ACGRP_NM" VARCHAR(400) variable not null,
    "DRCR_CD" VARCHAR(20) fixed not null,
    "ACGRP_LV" NUMERIC(3),
    "UP_GRP_CD" VARCHAR(80) variable,
    "ACGRP_FG" VARCHAR(20) fixed not null,
    "PRINT_NO" VARCHAR(160) variable default '1' not null
);
create table "CA_COA_LDTL"
(
    "GAAP_CD" VARCHAR(20) fixed not null,
    "ACCT_CD" VARCHAR(80) variable not null,

```

```

"LANG_CD" VARCHAR(20) fixed not null,
"ACCT_NM" VARCHAR(400) variable not null
);
WITH
T_ACCTAMT AS (
    SELECT      GRP.ACGRP_CD AS ACCT_CD,
                COALESCE(LANG.ACGRP_NM, GRP.ACGRP_NM) AS ACCT_NM,
                GRP.UP_GRP_CD AS UP_ACCT_CD,
                GRP.ACGRP_CD,
                GRP.DRCR_CD,
                GRP.ACGRP_FG,
                GRP.ACGRP_LV,
                NULL AS ACCT_LV,
                GRP.PRINT_NO AS PRINT_NO_GRP,
                GRP.PRINT_NO AS PRINT_NO_ACCT,
                0 AS DR_AMT_THIS
    FROM        CA_ACGRP_MST GRP
                LEFT OUTER JOIN CA_ACGRP_LDTL LANG
                    ON LANG.GAAP_CD = GRP.GAAP_CD
                    AND LANG.FORM_CD = GRP.FORM_CD
                    AND LANG.ACGRP_CD = GRP.ACGRP_CD
                    AND LANG.LANG_CD = 'KO'
    WHERE       GRP.ACGRP_LV != '1'
                AND GRP.GAAP_CD = '1'
                AND GRP.FORM_CD = 'D0012'
                AND GRP.ACGRP_FG != '999'
    UNION ALL
    SELECT      COA.ACCT_CD,
                NULL AS ACCT_NM,
                COALESCE(COA.UP_ACCT_CD, DTL.ACGRP_CD) AS UP_ACCT_CD,
                MST.ACGRP_CD,
                COA.DRCR_CD,
                COA.ACCT_FG,
                NULL AS ACGRP_LV,
                COA.ACCT_LV,
                MST.PRINT_NO AS PRINT_NO_GRP,
                COA.PRINT_NO AS PRINT_NO_ACCT,
                NULL
    FROM        CA_COA_MST COA
                INNER JOIN CA_ACGRP_DTL DTL
                    ON DTL.GAAP_CD = COA.GAAP_CD
                    AND DTL.ACCT_CD = COA.ROOT_ACCT_CD
                    AND DTL.FORM_CD = 'D0012'
                INNER JOIN CA_ACGRP_MST MST
                    ON MST.GAAP_CD = DTL.GAAP_CD
                    AND MST.FORM_CD = DTL.FORM_CD
                    AND MST.ACGRP_CD = DTL.ACGRP_CD
    WHERE       COA.GAAP_CD = '1'
                AND COA.ACCT_LV = '1'
),
T_SUMAMT(ACCT_CD, UP_ACCT_CD, DR_AMT_THIS) AS (

```

```

SELECT    ACCT_CD,
          UP_ACCT_CD,
          DR_AMT_THIS
FROM      T_ACCTAMT
WHERE     ACCT_LV IS NOT NULL
UNION ALL
SELECT    PR.ACCT_CD,
          PR.UP_ACCT_CD,
          CH.DR_AMT_THIS
FROM      T_ACCTAMT PR
          INNER JOIN T_SUMAMT CH
                ON CH.UP_ACCT_CD = PR.ACCT_CD
),
T_BASE_DATA AS (
  SELECT    COA.ACCT_CD,
            COA.ACCT_NM,
            COA.UP_ACCT_CD,
            COA.ACGRP_CD,
            COA.DRCR_CD,
            COA.ACGRP_FG,
            COA.ACGRP_LV,
            COA.ACCT_LV,
            COA.PRINT_NO_GRP,
            COA.PRINT_NO_ACCT,
            CASE WHEN COA.DRCR_CD = '1' THEN COALESCE(AMT.DR_AMT_THIS, 0)
                  ELSE COALESCE(AMT.DR_AMT_THIS, 0) - COALESCE(AMT.DR_AMT_THIS, 0)
            END AS DOCU_AMT_THIS,
            CASE WHEN COA.DRCR_CD = '1' THEN COALESCE(AMT.DR_AMT_THIS, 0)
                  ELSE COALESCE(AMT.DR_AMT_THIS, 0) - COALESCE(AMT.DR_AMT_THIS, 0)
            END AS DOCU_AMT_PREV
FROM      T_ACCTAMT COA
          LEFT OUTER JOIN (
            SELECT    ACCT_CD,
                      SUM(DR_AMT_THIS) AS DR_AMT_THIS
            FROM      T_SUMAMT
            GROUP BY  ACCT_CD
          ) AMT
          ON AMT.ACCT_CD = COA.ACCT_CD
),
T_NET_INCOME AS (
  SELECT    SUM(
            CASE WHEN (SELECT DRCR_CD FROM CA_ACGRP_MST WHERE ACGRP_FG = '842' AND
                      ELSE DOCU_AMT_THIS * (-1)
            END
          ) AS DOCU_AMT_THIS,
          SUM(
            CASE WHEN (SELECT DRCR_CD FROM CA_ACGRP_MST WHERE ACGRP_FG = '842' AND
                      ELSE DOCU_AMT_PREV * (-1)
            END
          ) AS DOCU_AMT_PREV
FROM      T_BASE_DATA BD

```

```

        INNER JOIN CA_ACGRP_MST MST
            ON MST.ACGRP_CD = BD.ACCT_CD
            AND MST.GAAP_CD = '1'
            AND MST.FORM_CD = 'D0032'
            AND MST.ACGRP_LV = '2'
    ),
    T_CAP_ACCT(ACCT_CD, UP_ACCT_CD) AS (
        SELECT      ACCT_CD,
                    UP_ACCT_CD
        FROM        T_ACCTAMT
        WHERE       ACGRP_FG = '821'
    UNION ALL
        SELECT      A.ACCT_CD,
                    A.UP_ACCT_CD
        FROM        T_ACCTAMT A
        INNER JOIN  T_CAP_ACCT B
            ON B.ACCT_CD = A.UP_ACCT_CD
    ),
    T_GROUP_B AS (
        SELECT      'GROUP_B' AS GROUP_CODE
        FROM        T_BASE_DATA BASE
        INNER JOIN  CA_ACGRP_MST MST
            ON MST.ACGRP_CD = BASE.ACCT_CD
            AND MST.GAAP_CD = '1'
            AND MST.FORM_CD = 'D0022'
            AND MST.ACGRP_LV = '2'
    ),
    T_GROUP_C AS (
        SELECT      'GROUP_C' AS GROUP_CODE
        FROM        (
            SELECT      ACCT_CD,
                        CASE WHEN ACGRP_FG = '821'
                            OR ACCT_CD = (SELECT MIN(ACCT_CD) FROM T_BASE_DATA )
                            THEN DOCU_AMT_THIS + (SELECT DOCU_AMT_THIS FROM T_NET_I
                                ELSE DOCU_AMT_THIS
                            END AS DOCU_AMT
            FROM        T_BASE_DATA
            WHERE       ACCT_CD IN (
                SELECT      ACCT_CD
                FROM        T_CAP_ACCT
            )
            AND        (
                ACCT_LV IS NULL
                OR (
                    ACCT_LV IS NOT NULL
                    AND UP_ACCT_CD = (SELECT MAX(ACGRP_CD) FROM CA_ACGRP_
                )
            )
        )
    UNION ALL
        SELECT      CONCAT(ACCT_CD, '1') AS ACCT_CD,
                    DOCU_AMT_THIS

```

```

        FROM      T_BASE_DATA
        WHERE      ACCT_CD = (
                    SELECT      MIN(ACCT_CD)
                    FROM        T_BASE_DATA
                    WHERE        ACGRP_FG = '826'
                    AND         ACCT_CD IN (SELECT ACCT_CD FROM T_CAP_ACCT)
                )
        UNION ALL
        SELECT      CONCAT(ACCT_CD, '2') AS ACCT_CD,
                    (SELECT DOCU_AMT_THIS FROM T_NET_INCOME) AS DOCU_AMT
        FROM        T_BASE_DATA
        WHERE      ACCT_CD = (
                    SELECT      MIN(ACCT_CD)
                    FROM        T_BASE_DATA
                    WHERE        ACGRP_FG = '826'
                    AND         ACCT_CD IN (SELECT ACCT_CD FROM T_CAP_ACCT)
                )
    )
    SELECT      GROUP_CODE
    FROM        T_GROUP_B
    UNION ALL
    SELECT      GROUP_CODE
    FROM        T_GROUP_C;

```

#### ◦ 수행 결과

[ERR-31455 : Failed to work because an internal exception occurred from an OS.[Contact .

#### ◦ 예상 결과

```

GROUP_CODE
-----
No rows selected.

```

#### • Workaround

```
alter system set __OPTIMIZER_VIEW_TARGET_ENABLE = 0;
```

#### • 변경사항

- Performance view
- Property
- Compile Option
- Error Code

**BUG-47806 sm cursor interface 에서 인자 정합성 검사.**

- **module** : sm\_interface
- **Category** : Fatal
- **재현 빈도** : Rare
- **증상** : 내부적으로 sm cursor Interface 에서 파라미터의 null pointer에 대한 예외처리가 누락되어 있어, 비정상 종료 할 수 있습니다. 예외처리를 보강하여, 비정상 종료 하지 않도록 수정합니다.
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## BUG-47807 계층형 쿼리에 join을 같이 사용하고 start with구문에 level 컬럼이 사용된 경우 비정상 종료할 수 있습니다.

- **module** : qp
- **Category** : Fatal
- **재현 빈도** : Always
- **증상** : 계층형 쿼리에 join을 같이사용하고 start with구문에 level 컬럼이 사용된 경우 optimize에서 비정상 종료할 수 있어 수정합니다.
- **재현 방법**
  - 재현 절차

```
SELECT LEVEL
FROM DUAL A,
      (SELECT level start_level, level end_level
       FROM dual) B
WHERE 1=1
START WITH LEVEL >= B.START_LEVEL
CONNECT BY LEVEL >= B.START_LEVEL AND LEVEL <= B.END_LEVEL;
```

### ◦ 수행 결과

[ERR-31455 : Failed to work because an internal exception occurred from an OS.[Contact .

### ◦ 예상 결과



Not exception

- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## **BUG-47813 PCB를 LRU List에 등록한후 PCB에 접근하면 동시성 문제가 발생할 수 있습니다.**

- **module** : mm-planccache
- **Category** : Assert
- **재현 빈도** : Impossible
- **증상** : PCB(Plan Cache Block)를 LRU List에 등록한 후 PCB에 접근하는 경우, 잘못된 메모리를 참조하여 비정상 종료가 발생할 수 있습니다. LRU List에 PCB를 등록한 후에는 PCB에 Thread가 동시 접근하지 못하도록 수정합니다.
- **재현 방법**
  - 재현 절차
  - 수행 결과
  - 예상 결과
- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## **BUG-47820 start with 구문에 level pseudo column이 사용될 경우 결과값이 틀립니다.**

- **module** : qp
- **Category** : Functional Error
- **재현 빈도** : Always
- **증상** : start with 구문에 level pseudo column이 사용될 경우 결과오류가 발생하여 수정합니다.
- **재현 방법**
  - 재현 절차

```

drop table t1;
create table t1 ( i1 integer , i2 integer );
insert into t1 values ( 0,2 );
insert into t1 values ( 1,2 );
SELECT LEVEL
      FROM ( select C.start_level, C.end_level from DUAL A, (SELECT  i1 start_level, i2 en
                    FROM  t1  ) C ) B
 WHERE 1=1
    START WITH LEVEL >= B.START_LEVEL
    CONNECT BY LEVEL >= B.START_LEVEL AND LEVEL <= B.END_LEVEL;

```

#### ◦ 수행 결과

```

LEVEL
-----
1
2
2
1
2
2
6 rows selected.

```

#### ◦ 예상 결과

```

LEVEL
-----
1
2
2
3 rows selected.

```

- **Workaround**
- **변경사항**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## Changes

## Version Info

---

altibase version	database binary version	meta version	cm protocol version	replication protocol version	sharding version
7.1.0.3.8	6.5.1	8.7.1	7.1.7	7.4.5	2.2.1

Altibase 7.1 패치 버전별 히스토리는 [Version\\_Histories](#) 에서 확인할 수 있다.

## 호환성

### Database binary version

데이터베이스 바이너리 버전은 변경되지 않았다.

데이터베이스 바이너리 버전은 데이터베이스 이미지 파일과 로그파일의 호환성을 나타낸다. 이 버전이 다른 경우의 패치(업그레이드 포함)는 데이터베이스를 재구성해야 한다.

### Meta Version

메타 버전은 변경되지 않았다.

패치를 롤백하려는 경우,  
[메타다운그레이드](#)를  
참고한다.

### CM protocol Version

통신 프로토콜 버전은 변경되지 않았다.

### Replication protocol Version

Replication 프로토콜 버전은 변경되지 않았다.

### Sharding Version

샤딩 버전은 변경 되지 않았다.

알티베이스 샤딩 프로토콜 및 메타는 상위, 하위 호환성을 보장하지 않는다. 즉, 샤딩 버전이 다른 경우, 재구성해야 한다.

## 프로퍼티

추가/변경/삭제된 프로퍼티 없음

# 성능 뷰

추가/변경/삭제된 성능뷰 없음